

```

1  from numpy import linspace, zeros, log, exp
2  from matplotlib.pyplot import plot, xscale, yscale
3
4  def u(x) :
5      return exp(-x)
6
7  def x(xi,a,c,m):
8      return a + c*xi/(1 - xi)**m
9
10 def dxdxi(xi,c,m):
11     return c*(1 + (m - 1)*xi)/(1 - xi)**(m+1)
12
13 def Integration(u,a,c,m,N) :
14     integral = 0.
15     for n in range(1,N+1) :
16         xi = (n - 1/2)/N
17         integral = integral + u(x(xi,a,c,m))*dxdxi(xi,c,m)*(1/N)
18     return integral
19
20 a = 0.; c = 1.; m = 1.
21
22 N = 1
23 r = 2; S = 5
24 p = 2; q = 2
25
26 U = zeros((S,S))
27 R = zeros((S,S))
28 p_eff = zeros((S,S))
29
30 for s in range(S) :
31     U[s,0] = Integration(u,a,c,m,r**s*N)
32
33 for s in range(1,S) :
34     for l in range(s) :
35         R[s,l] = (U[s,l] - U[s-1,l])/(r**(p + l*q) - 1)
36         U[s,l+1] = U[s,l] + R[s,l]
37
38 for s in range(2,S) :
39     for l in range(s-1) :
40         p_eff[s,l] = log(abs(R[s-1,l]/R[s,l]))/log(r)
41
42 # Функция выводит форматированную таблицу
43 def PrintTriangular(A,i) :
44     print('      ',end=' ')
45     for l in range(len(A)) :
46         print(' p={0:<4d}'.format(p + l*q),end=' ')
47     print()
48     for m in range(len(A)) :
49         print(' s={0:<2d}'.format(m),end=' ')
50         for l in range(m + 1 - i) :
51             print('{0:7.4f}'.format(A[m,l]),end=' ')
52     print()
53     print()
54
55 print('Таблица приближённых значений интеграла:')
56 PrintTriangular(U,0)
57 print('Таблица оценок ошибок:')
58 PrintTriangular(R,1)
59 print('Таблица эффективных порядков точности:')
60 PrintTriangular(p_eff,2)
61
62 plot([r**s*N for s in range(1,S)],abs(R[1:,0]),'-bo')
63 xscale('log'); yscale('log')
64

```

```
65 # Листинг программы,  
66 # реализующей приближённое вычисление несобственного интеграла  
67 # с помощью рекуррентного сгущения сеток и многократного повышения  
68 # точности по Ричардсону (с вычислением эффективных порядков точности)
```